Express Mail Label No. EV303487090US

COMPUTER SYSTEM CACHE CONTROLLER AND METHODS OF OPERATION OF A CACHE CONTROLLER

BACKGROUND

[0001] Cache memories and cache controllers may be used on computer systems to improve latency of accessing frequently used data. A cache controller may be placed between a chipset (*e.g.,* a chipset comprising a North bridge, South bridge and possibly other peripheral integrated circuits) and one or more processors (*e.g.,* CPUs). In many computer systems, cache memories may contain a duplicate copy of a portion of a computer's main memory and may be organized in multiple levels (*e.g.,* level one, level two, level three, and level four) according to size and/or proximity to the processor. For example, a level one ("L1") cache memory may be closest to a processor (*e.g.,* on the same substrate as the processor) and may implement a small amount of high performance memory that stores the most commonly requested data. If the processor cannot find the requested data in the L1 cache, the level two ("L2") cache, which may contain a larger segment of memory, may be accessed. The process of searching the next largest cache level for the requested data may continue until the data is found. If the data is not found in one of the cache memories, the main memory may be accessed.

[0002] An increasing number of computer systems implement processors and chipsets with different communication protocols and/or physical interfaces. Cache controllers, coupled between the chipset and processor, in these mixed protocol systems may be specifically designed for one interface to communicate with a chipset using a particular protocol, and for a second interface to communicate with the processor using a different protocol. Should any one

parameter change (*i.e.*, the device with which an interface communicates or the communication protocol), another specifically designed cache controller may be required.

SUMMARY

[0003] The problems noted above are solved in large part by a cache controller that allows communication between a CPU and a bridge device using different communication protocols. At least some embodiments may be a computer system comprising a central processing unit ("CPU"), a bridge device coupled to a main memory, and a cache controller coupled between the bridge device and the CPU. The computer system further comprises a cache memory coupled to the cache controller and providing cache memory space to the CPU. The cache controller allows communication between the CPU and the bridge device when the CPU communicates using a first protocol and the bridge device communicates using a second protocol. The cache controller also allows communication between the CPU and the bridge device when the CPU communicates using the second protocol and the bridge device communicates using the first protocol.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] For a detailed description of exemplary embodiments of the invention, reference will now be made to the accompanying drawings in which:

[0005] Figure 1 illustrates a block diagram of a system in accordance with embodiments of the invention;

[0006] Figures 2 illustrates a block diagram of a system in accordance with alternative embodiments of the invention;

[0007] Figure 3 illustrates a cache controller in accordance with embodiments of the invention; and

[0008] Figure 4 illustrates a cache control method in accordance with embodiments of the invention.

NOTATION AND NOMENCLATURE

[0009] Certain terms are used throughout the following description and claims to refer to particular system components. As one skilled in the art will appreciate, computer companies may refer to a component by different names. This

document does not intend to distinguish between components that differ in name but not function.

[0010] In the following discussion and in the claims, the terms "including" and "comprising" are used in an open-ended fashion, and thus should be interpreted to mean "including, but not limited to... ." Also, the term "couple" or "couples" is intended to mean either an indirect or direct electrical connection. Thus, if a first device couples to a second device, that connection may be through a direct electrical connection, or through an indirect electrical connection via other devices and connections.

DETAILED DESCRIPTION

[0011] The following discussion is directed to various embodiments of the invention. Although one or more of these embodiments may be preferred, the embodiments disclosed should not be interpreted, or otherwise used, as limiting the scope of the disclosure. In addition, one skilled in the art will understand that the following description has broad application, and the discussion of any embodiment is meant only to be exemplary of that embodiment, and not intended to intimate that the scope of the disclosure is limited to that embodiment.

[0012] Figure 1 illustrates a computer system 100 in accordance with embodiments of the invention. As shown in Figure 1, the computer system 100 may comprise a cache controller 110 coupled to a chipset 102, a cache memory 108, and a processor ("CPU") 120. The chipset 102, which may comprise a North bridge and a South bridge, may couple the CPU 120 to a main memory 104, an input/output ("I/O") device 106, and a storage device 107. The chipset 102 and the CPU 120 may implement differing communication protocols and differing bus architectures (*e.g.,* the architectures may differ in bus width and/or clock speed). For example, the chipset 102 may be an Intel Architecture ("IA") chipset (*e.g.,* IA-32 chipset) that implements an IA protocol (*e.g.,* IA-32). The CPU 120 may be an Itanium processor family ("IPF") processor which implements an IPF protocol. Alternatively, the chipset 102 may be an Itanium processor family ("IPF") chipset and the CPU 120 may be an IA processor. The various protocols presented are merely exemplary, other protocols may be equivalently used.

[0013] Using a communication protocol (such as an IA-32 or IPF protocol) the CPU 120 may access computer readable instructions and/or data (hereinafter collectively or in the alternative just "data") stored in the main memory 104 and the cache memory 108. In accordance with at least some embodiments, the cache memory 108 may store duplicate copies of data such that the CPU 120 may more quickly locate and/or access the data.

[0014] In accordance with at least some embodiments, the cache memory 108 may be implemented as a level four ("L4") cache for use with the CPU 120. Accordingly, the CPU 120 may have access to other caches (*e.g.,* cache levels 1-3) which are not shown in Figure 1. In such embodiments, if a CPU request is not fulfilled by the lower level caches, that CPU request may be forwarded to and processed by the cache controller 110 of the cache memory 108.

[0015] By storing data in the cache memory 108, the CPU 120 may access the data without accessing the main memory 104 or other storage mediums. This may provide faster memory accesses for several reasons. For example, in at least some embodiments, the main memory 104 or other storage mediums (hereafter main memory 104) may be significantly larger (*i.e.,* includes more addresses) than the cache memory 108, whereby locating and accessing a particular address takes longer. Additionally, the main memory 104 may be a lower performance memory (*i.e.,* the clocking rate is lower and/or the time it takes to read from and/or write to an address may be longer). Additionally, accessing the main memory 104 through the chipset 102 may increase the time required for memory accesses by adding distance and/or logic between the CPU 120 and the main memory 104. Thus, accessing data stored in the cache memory 108 may enhance the performance of the computer system 100.

[0016] Data stored in the cache memory 108 may be organized and/or accessed as directed by the cache controller 110. The cache controller 110 controls the data stored and accessed in the cache memory 108 using a cache memory interface 116 coupled to the cache memory 108 and switch logic 111. The switch logic 111 also may couple a first protocol interface 112 to a second protocol interface 114. Accordingly, the switch logic 111 may direct data between

the cache memory interface 116, the first protocol interface 112, and the second protocol interface 114 as will later be described.

[0017] In the exemplary system 100 of Figure 1, the CPU 120 may send a request for data stored at an address location to the cache controller 110 using the second protocol interface 114. The second protocol interface 114 may comprise physical electrical signal lines and logic that interprets data received on the physical electrical signal lines (*i.e.,* a bus interface). Accordingly, the second protocol interface 114 may receive and /or format data using a bus width, data transfer rate, data frame formatting and/or data command encoding according to the protocol used by the CPU 120. The request sent by the CPU 120 may be forwarded from the second protocol interface 114 to the switch logic 111. The communication protocol used between the switch logic 111 and the interfaces 112, 114 and 116 need not necessarily be the same as the communication protocols used by the interfaces 112 and 114 when communicating to external devices (*e.g.,* the CPU 120 and the chipset 102).

[0018] The switch logic 111 sends the request from the CPU 120 to the cache memory interface 116, which controls reading and writing accesses to the cache memory 108. The cache memory interface 116 compares the request (*e.g.,* an address location) with address locations stored in a tag memory 118 coupled to the cache memory interface 116. For example, the tag memory 118 may store full or partial main memory address locations of data that is also stored in the cache memory 108. If tag information (*e.g.,* address information) in the tag memory 118 matches the request provided by the CPU 120, referred to as a "cache hit," the cache memory interface 116 may access the requested data from the cache memory 108. The requested data may then be sent to the CPU 120 through the switch logic 111 and the second protocol interface 114. If tag information in the tag memory 118 does not match the information sent by the CPU 120, referred to as a "cache miss," the cache memory interface 116 may forward the request to the chipset 102 through the switch logic 111 and the first protocol interface 112, whereby the main memory 104, the I/O device 106, or the storage device 107 may be accessed to provide the requested data.

[0019] To forward the request from the cache controller 110 to the chipset 102, the first protocol interface 112 may prepare the information according to a protocol that differs from the protocol implemented by the CPU 120 such that the chipset 102 may correctly interpret and use the request. Accordingly, the first protocol interface 112 may comprise physical electrical signal lines and logic that interprets/formats data received on the physical electrical signal lines (*i.e.,* a bus interface). Accordingly, the first protocol interface 112 may receive and/or format data using a bus width, data transfer rate, data frame formatting and/or data command encoding according to the protocol used by the chipset 102. For example, the first protocol interface 112 may format requests from an IA-32 protocol into an IPF protocol or vice versa. The chipset 102 may use the request from the CPU 120 to access an address location in the main memory 104 whereby data associated with the address location is returned to the CPU 120 through the cache controller 110.

[0020] After the chipset returns the data following a cache miss, the cache controller 110 may store the data in the cache memory 108, and also store tag information in the tag memory 118 such that a future request for the same address will be a cache hit. Finally, the cache controller 110 may forward the requested data to the CPU 120 through the switch logic 111 and the second protocol interface 114. In updating the cache memory, the cache controller 110 may overwrite cache entries containing data that have not been requested recently.

[0021] The interfaces 112 and 114 of a cache controller in accordance with embodiments of the invention, however, are not constrained to communication with just one type of hardware (*e.g.*, just a chipset or just a CPU). Figure 2 illustrates a system 200 in accordance with alternative embodiments of the invention where the cache controller 110 couples to the chipset using the second communication protocol (unlike Figure 1 where the chipset uses the first communication protocol), and the cache controller 110 also couples to the CPU using the first communication protocol (unlike Figure 1 where the CPU uses the second communication protocol). Thus, the system 200 illustrates the ability of the cache controller 110 to function with "opposite" or "reversed" CPU/chipset

combinations. Stated otherwise, the cache controller 110 may communicate with either a CPU that implements a first protocol and a chipset that implements a second protocol, or the cache controller 110 may communicate with a CPU that implements a second protocol and a chipset that implements a first protocol.

[0022] Figure 3 illustrates a cache controller 150 in accordance with alternative embodiments of the invention. The cache controller 150 may be configurable to couple to a cache memory, one or more processors, and a chipset. The function of the cache controller 150 is similar in function to the cache controller 110 described for Figures 1 and 2 but with the ability to interface a greater variety of processor and chipset combinations. Accordingly, the cache controller 150 may comprise a plurality of first protocol interfaces 112A, 112B and a plurality of second protocol interfaces 114A, 114B coupled to the logic switch 111.

[0023] The cache controller 150 may couple between a variety of processor and chipset combinations such as those illustrated in the Table 1 shown below.

Cache controller (150) configurations or modes	Processor	Chipset
1	First protocol	First protocol
2	First protocol	Second protocol
3	Second protocol	First protocol
4	Second protocol	Second protocol

Table 1

[0024] As shown in Table 1, the cache controller 150 may be configured for use in a number of modes in accordance with any combination of chipsets and processors associated with a first and a second protocol. As previously explained, the first and second communications protocols may be an IA-32 protocol and an IPF protocol, respectively. However, the first and second protocols may be selected from any number of protocols now known (*e.g.,* IA-32, IPF, Hyper Transport, PowerPC) or later invented.

[0025] In at least some embodiments, a user may configure the cache controller 150 to operate in a particular mode using one or more control signals that couple to the control module 152. For example, by applying a predetermined voltage level for the control signals, a user may configure the cache controller 150 for use with one of the processor/chipset combination modes described above.

[0026] The cache controllers 110 and 150 may be implemented as integrated circuits packaged as chips. Therefore, a user may configure a cache controller (*e.g.,* cache controller 110, 150) for use with a particular processor/chipset combination by applying a voltage to (or grounding) one or more predetermined pins of the chip.

[0027] Figure 4 illustrates a method 400 in accordance with embodiments of the invention. As shown in Figure 4, the method 400 may start (block 401) and move to configuring a first port (*i.e.,* interface) of a cache controller for use with a device that implements a first protocol (block 402). For example, the first port may be configured for use with processors or chipsets that implement a first protocol as described above. A second port (*i.e.,* interface) of the cache controller may be configured for use with a device that implements a second protocol (block 404), and thus the process 400 may end (block 406). For example, the second port may be configured for use with processors or chipsets that implement a second protocol as described above.

[0028] The above discussion is meant to be illustrative of the principles and various embodiments of the present invention. Numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. For example, a cache controller may be used with devices that implement more than two communication protocols. In such embodiments, the cache controller design may be adjusted to account for additional protocol interfaces such as those described above. It is intended that the following claims be interpreted to embrace all such variations and modifications.